A HEURISTIC APPROACH TO THE OPTIMIZATION OF CENTRALIZED COMMUNICATION NETWORKS

by

CARLOS R. De BACKER*

Working Paper 927-77 May, 1977

A HEURISTIC APPROACH TO THE OPTIMIZATION OF

CENTRALIZED COMMUNICATION NETWORKS

by

CARLOS R. De BACKER*

Working Paper 927-77 May, 1977

* Visiting Fellow from Universiteit Antwerpen
(UFSIA), Belgium.

A HEURISTIC APPROACH TO THE OPTIMIZATION OF CENTRALIZED COMMUNICATION NETWORKS

ABSTRACT

In this working paper a heuristic optimization technique for centralized communication networks will be described. The optimization procedure can be used to determine: (1) the topology of one-level networks without concentrators, (2) the topology, the number and locations of concentrators in two-level networks, and (3) the topology of loop networks. Beside the topology aspect, also the line capacities and line organization structures (polling/contention; priorities of input or output on half-duplex lines vs. full-duplex lines) are computed. By means of a feed-back technique the model guarantees a solution which corresponds to the specified average response time.

1. Introduction

The classification of computer communication networks into centralized and distributed networks depends on the locations of the computers and the data bases. In a centralized network all processing capabilities and data bases will be located at a central site. A distributed network, however, will have its processors and data bases spread over a set of locations. In this paper we will focus on the optimization of centralized teleprocessing networks. In a forthcoming paper by J. Akoka and P. Chen (1) an optimization technique for distributed systems will be presented. For a more detailed survey on the issue of distributed communications networks, see reference (2).

The problem of optimizing a centralized communication network can be viewed as minimizing the overall network cost, provided that the response time requirements at the various terminal sites are satisfied. Generally we are given the location of the CPU, the locations of the terminals, the cost of communication lines, the cost of concentrators, and traffic data from terminals and central computer. The objective is to find an optimal combination of the following design variables: (1) topology of the network, (2) line capacities, (3) number of concentrators, (4) location of concentrators, (5) line organization structures (polling or contention), such that the overall network cost is minimized and the response time requirement is not violated.

In view of the complexity of the problem it is practically impossible to use exact techniques for generating the mathematical optimum. So, a heuristic approach for solving the optimization of centralized communication networks will be studied in this paper. In contrast with partial solutions generated by various authors, this network procedure will cover all major design elements.

2. The overall network procedure

The flow-chart in Figure 1 illustrates the three basic steps in the overall network optimization procedure: optimization - analysis - evaluation.

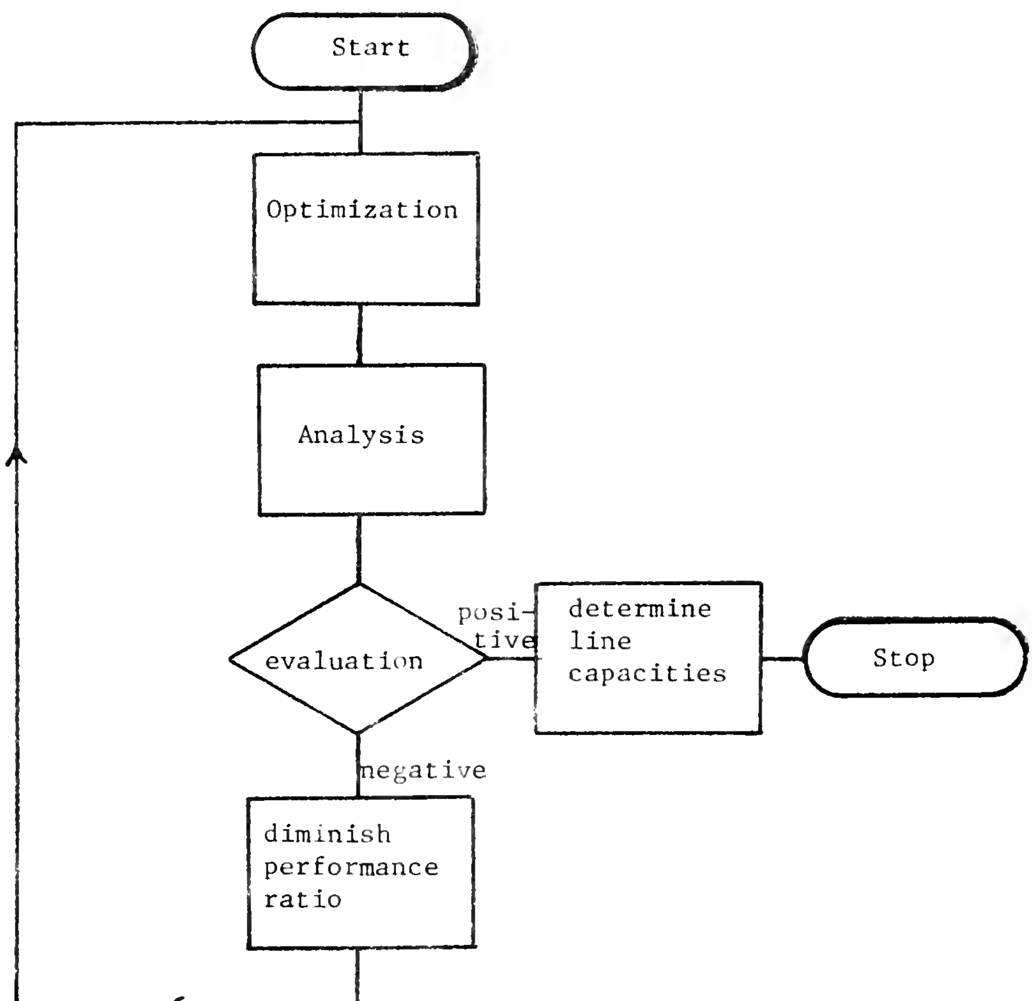


Figure 1: The overall network optimization procedure

In the optimization step the topology will be determined, given a fixed line capacity over the network. Three network types will be covered: one-level multidrop networks without concentrators; two-level networks with concentrators optimally located, and loop networks. The optimization heuristics for one- and two-level networks are based upon MST (Minimal Spanning Tree) type algorithms with inclusion of maximal performance constraints (line utilization).

The determination of the response times under several organization schemes is the subject of the analysis part. For contention systems basic queuing formulas under several traffic loads generally give good computation results. However, for polling systems either an extensive analytical model or simulation techniques have to be used.

In the evaluation section the computed response times are compared with the response time objectives. When the computed response time is lower than the objective, the line capacities of the high-level network (between computer and concentrators) can be computed. However, when the computed response time exceeds the objective, the optimization procedure has to be recomputed with a lower performance ratio.

3. Optimization

In a first section (3.1) of this paragraph optimization techniques for one-level multidrop networks without concentrators will be reviewed and evaluated. In a second section (3.2) we shall deal with optimization algorithms for networks with concentrators (the so-called two-level networks). In section 3.3 a heuristic algorithm for loop networks will be presented.

3.1 One-level networks without concentrators

In the early days of teleprocessing, terminals were star connected (via point-to-point lines) to the CPU. Since this was not the most economical way of connecting a set of terminals to a single device, and because of the enormous waste of computer ports, network designers built multidrop networks using only one single channel for a set of terminals.

The problem of optimizing a multidrop network has been investigated by many authors. Chandy and Lo (3) describe an exact method using the branch and bound algorithm. This technique however, is only useful for the solution of small size problems.

Heuristic methods for solving the multidrop optimization problem can be based upon MST (Minimal Spanning Tree) algorithms, adapted to include a performance ratio (in our case a maximum utilization rate of a link). Some of the major heuristics will be reviewed briefly.

Only the basic steps of the algorithms will be reviewed. For a more detailed survey, see reference (6).

KRUSKAL'S ALGORITHM (7)

[1] Initially every terminal is treated as a separate component;

[2] Determine the minimum cost link between two components not violating the performance constraint;

[3] Join those two components. If all terminals are connected: end; otherwise return to [2].

Illustration: Figure 2.

Data for illustrations 2, 3, 4, 5

Cost Matrix

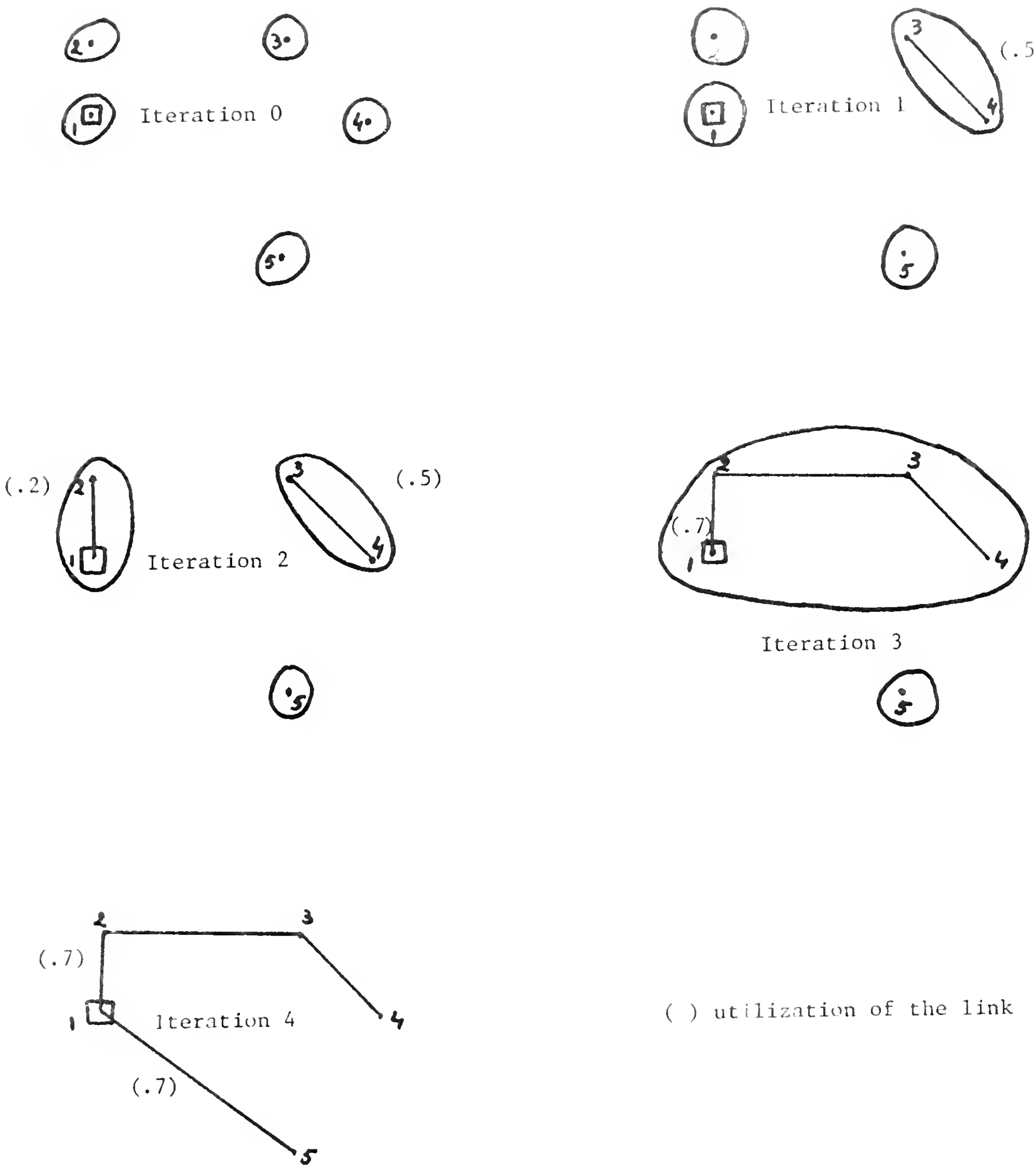
	1	2	3	4	5
1	-	10	26	27	30
2	10	-	25	32	37
3	26	25	-	8	23
4	27	32	8	-	22
5	30	37	23	22	-

Average number of bits from

2	40
3	20
4	80
5	140

Line Capacity 200 bits/sec.
Max. Line utilization: 0.7

Illustration: Figure 2: KRUSKAL



PRIM'S ALGORITHM (5)

[1] Initially {A} contains only the central node; {B} contains all other nodes;

[2] Find the minimum cost link between any node of {B} and any node of {A} not violating the performance constraint;

[3] Link the two nodes together. Remove the newly connected node out of {B} and put it in {A} ;

[4] If { B } is idle: end; otherwise return to [2].

Illustration: Figure 3.

ESAU/WILLIAMS' ALGORITHM (8)

[1] The initial configuration is a star-structure with the CPU location c as central node;

[2] Find two nodes i and j, not violating the performance constraint, and yielding the greatest cost savings when removing (i, c) and replacing it by (i, j);

[3] If this transformation does not exist: end; otherwise remove (i, c) and add (i, j); return to [2].

Illustration: Figure 4.

VAM ALGORITHM (9)

The VAM technique (VOGEL APPROXIMATION METHOD) is usually used in Operations Research for generating an initial solution for the Transportation Problem.

[1] Determine the trade-off values t_{ij}

$$t_{ij} = \underbrace{b_i - a_i}_{g_i} - c_{ij} = g_i - c_{ij}$$

where t_{ij} = trade-off value for link $i \rightarrow j$

a_i = the cheapest link leaving from node i

b_i = the second cheapest link leaving from node i

Illustration: Figure 3: PRIM

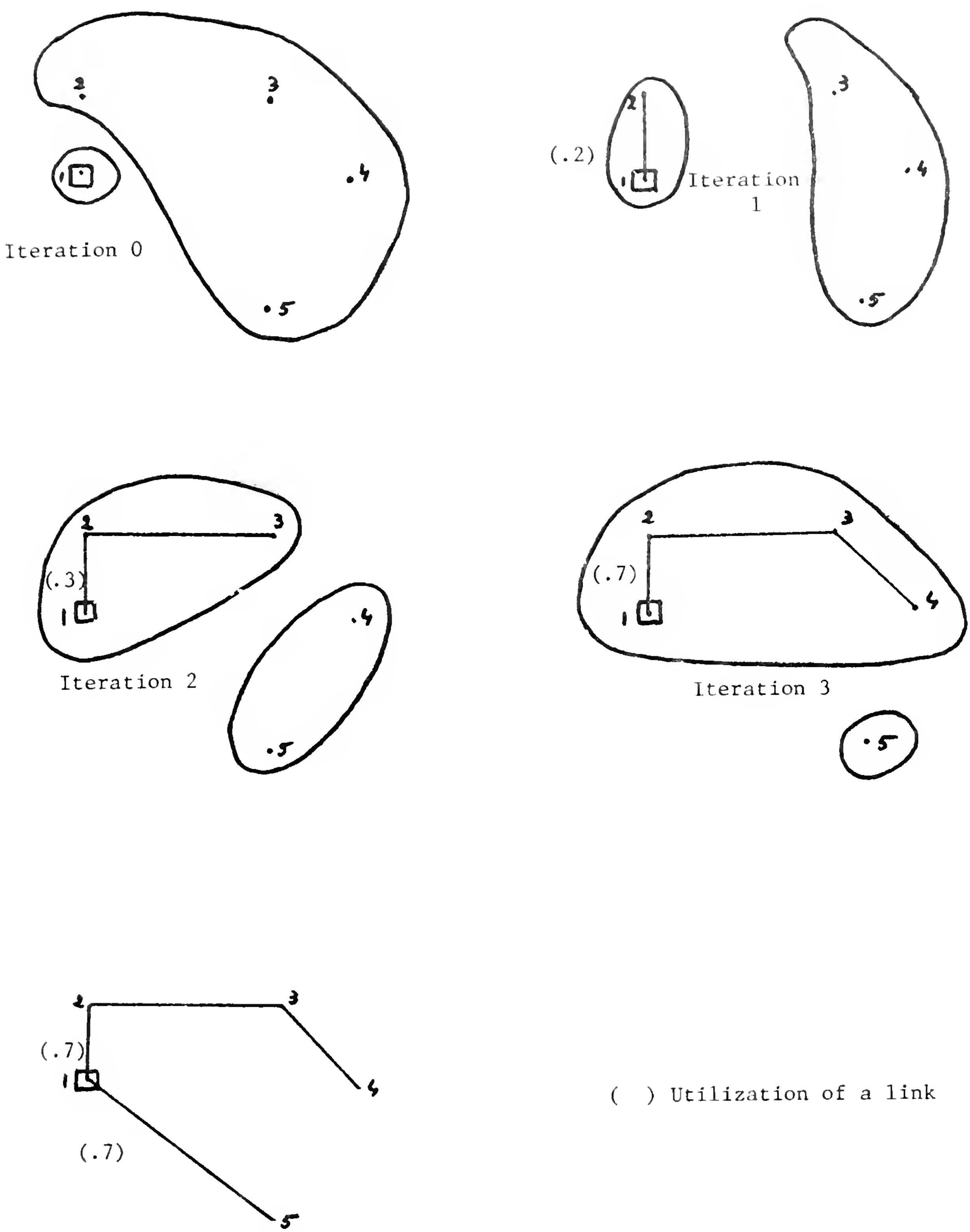


Illustration: Figure 4: ESAU-WILLIAMS

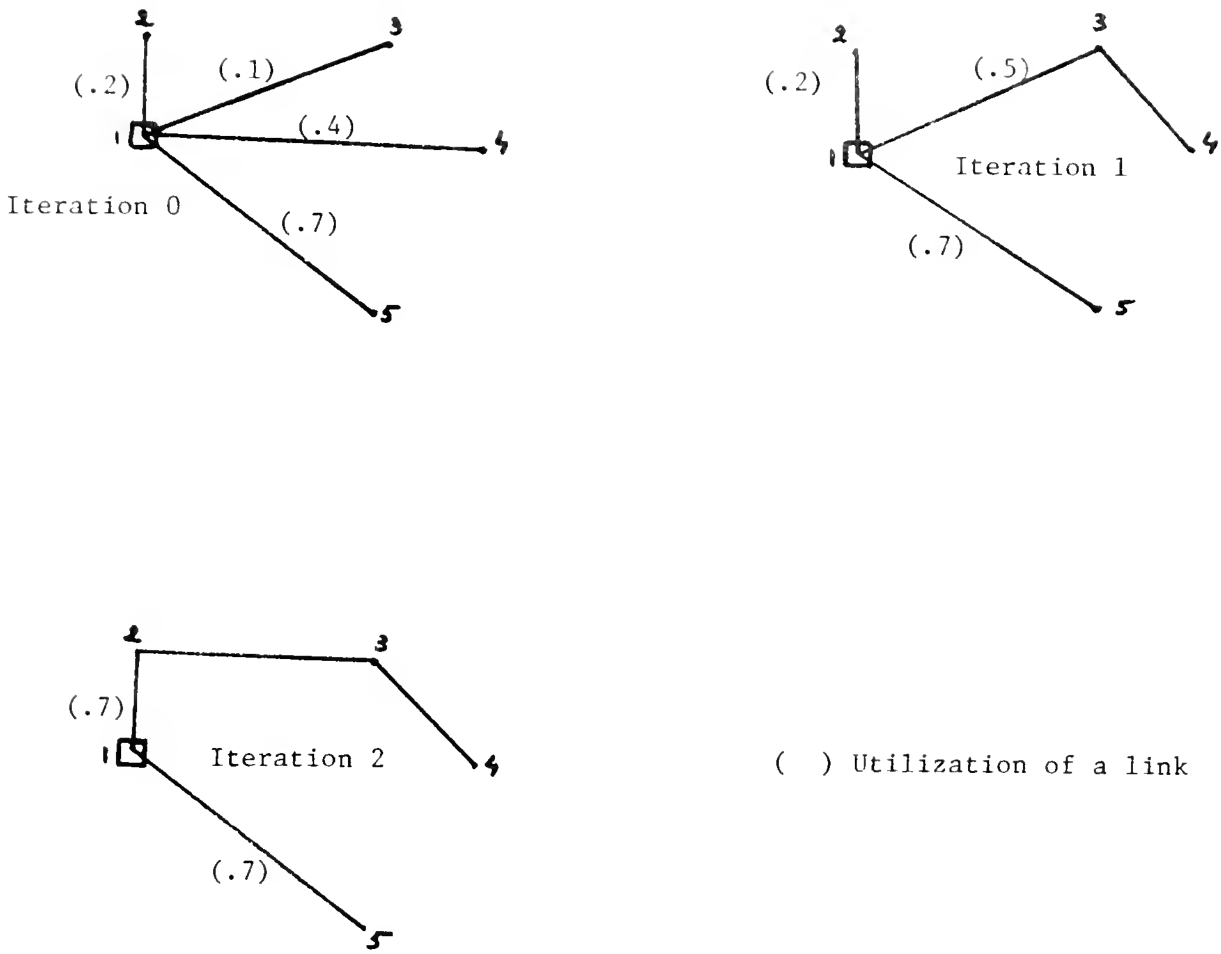
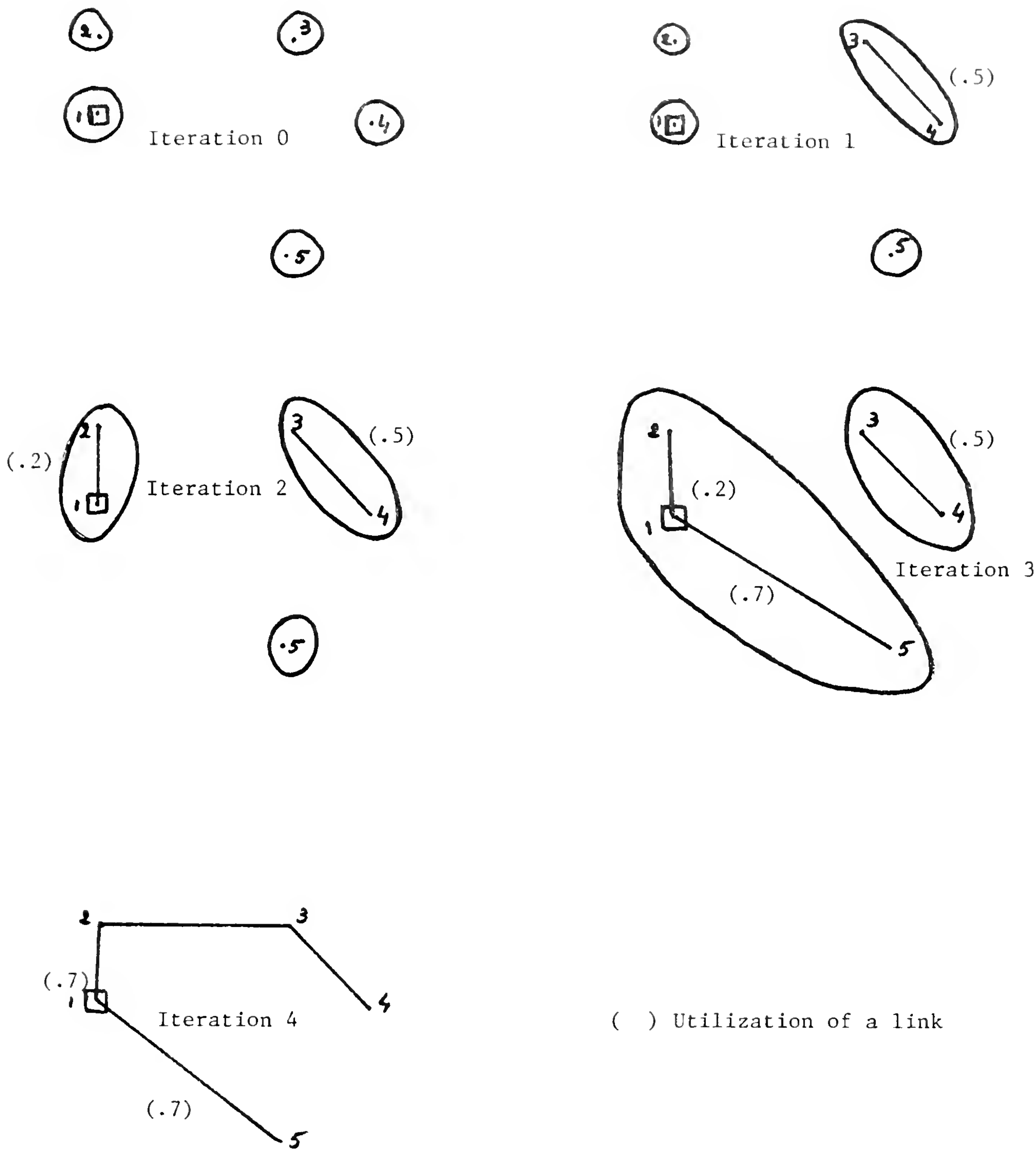


Illustration: Figure 5: VAM



$g_i \quad = \quad b_i - a_i$

$c_{ij} \quad =$ the cost of a communication line between node i and j

[2] Determine $t^*_{i^*j^*} = \max\limits_{i,j} t_{ij}$ not violating the performance constraint;

[3] Add link $i^* \rightarrow j^*$ to the network forming a new component;

[4] If all terminals are connected: end; otherwise return to [2].

Illustration: Figure 5.

COMPARISON

A first comparison has been made by executing the four algorithms on simulated data. The solutions for 100-node networks, with a maximum utilization rate of 0.7 are printed in Table I. The main results are summarized in Table II.

KRUSKAL		PRIM		ESAU		VAM	
2268	(2)	2425	(4)	1814	(1)	2412	(3)
1242	(2)	1722	(4)	1255	(3)	1189	(1)
1932	(4)	1229	(2)	1025	(1)	1248	(3)
1877	(2)	2780	(4)	1906	(3)	1725	(1)
3821	(3)	2923	(1)	3012	(2)	3821	(3)
1365	(1)	1365	(1)	1423	(2)	1365	(1)
2155	(2)	2289	(3)	1848	(1)	2155	(2)
1344	(2)	1584	(4)	1401	(3)	1246	(1)
1868	(2)	2140	(3)	2140	(3)	1740	(1)
1993	(2)	2295	(3)	1762	(1)	1762	(1)
2083	(3)	2083	(3)	2008	(2)	1833	(1)
2776	(3)	2807	(4)	2551	(2)	1932	(1)
1437	(1)	2131	(3)	1438	(2)	1437	(1)
2741	(2)	3084	(4)	2370	(1)	2998	(3)
2034	(2)	2478	(3)	2014	(1)	2034	(2)
1083	(1)	1200	(4)	1118	(2)	1124	(3)
2155	(2)	3245	(3)	1868	(1)	2155	(2)
2151	(3)	2067	(2)	2021	(1)	2151	(3)
1954	(1)	2581	(3)	2067	(2)	1954	(1)
2011	(1)	2055	(2)	2499	(3)	2011	(1)

( ) relative importance

TABLE I: Simulation results (in thousands)

TABLE 11: Survey of simulation results

relative importance	1	2	3	4
algorithm				
KRUSKAL	5	10	4	1
PRIM	2	3	8	7
ESAU	8	7	5	–
VAM	11	3	6	–

Another measure for comparing the four algorithms is the cost/line utilization graph. Two very representative graphs are depicted in Figures 6 and 7. Two major conclusions can be formulated:

(i) The maximum utilization rate strongly affects the performance of the algorithms. In Figure 7, e.g., VAM generates the best solution for ρ_{max} (maximum utilization rate) = 0.8. However, for ρ_{max} = 0.7, VAM generates the worst-case solution.

(ii) A stronger performance constraint can result in a network topology with a lower cost (varying ρ_{max} from 0.8 to 0.7 in Figure 6).

Since all algorithms may generate the best solution, the four heuristic techniques must be investigated during the design phase. A very flexible heuristic has been proposed in (6). The method deals with the remarks mentioned above. The flow-chart in Figure 8 illustrates the main steps of the design algorithm.

Figure 6

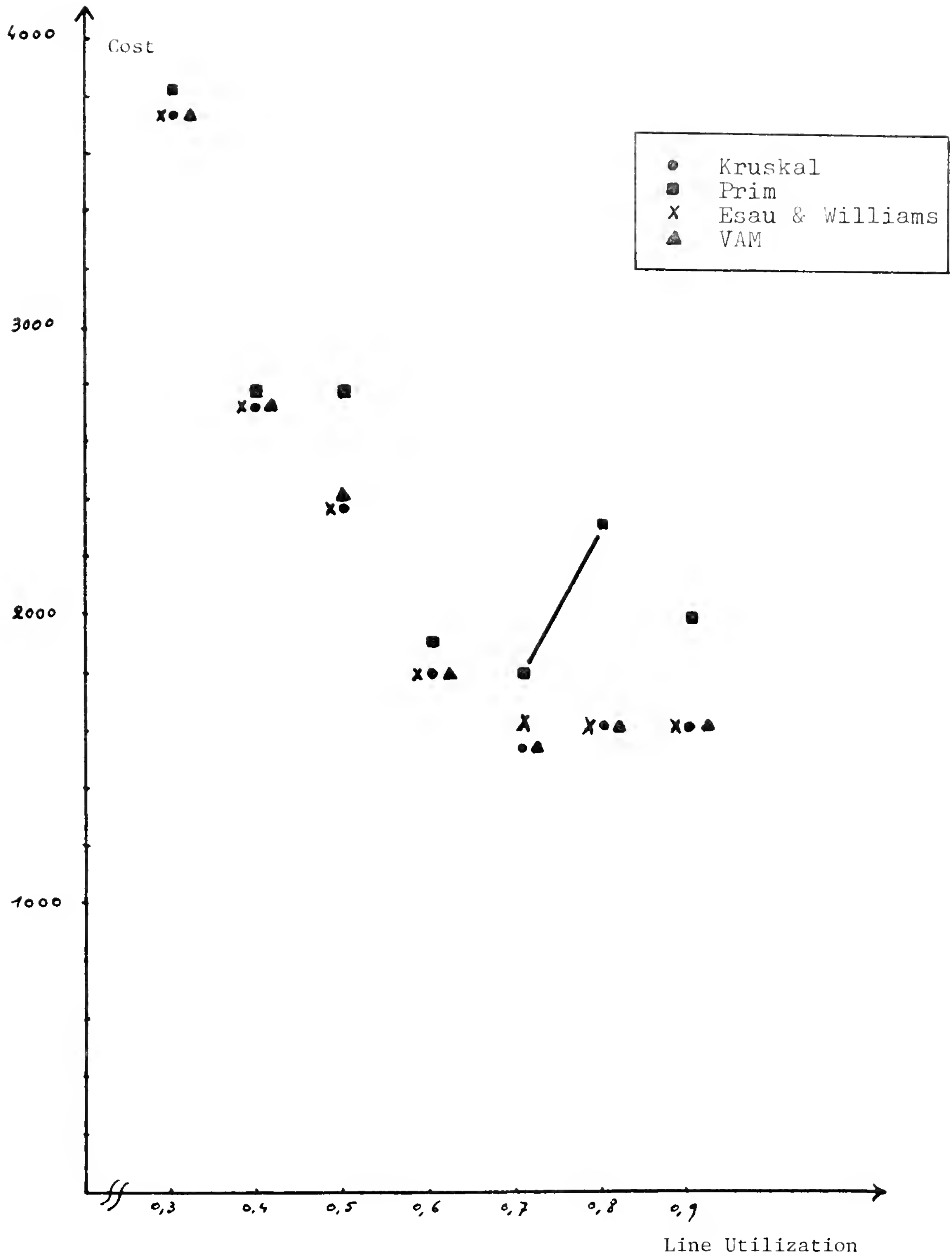
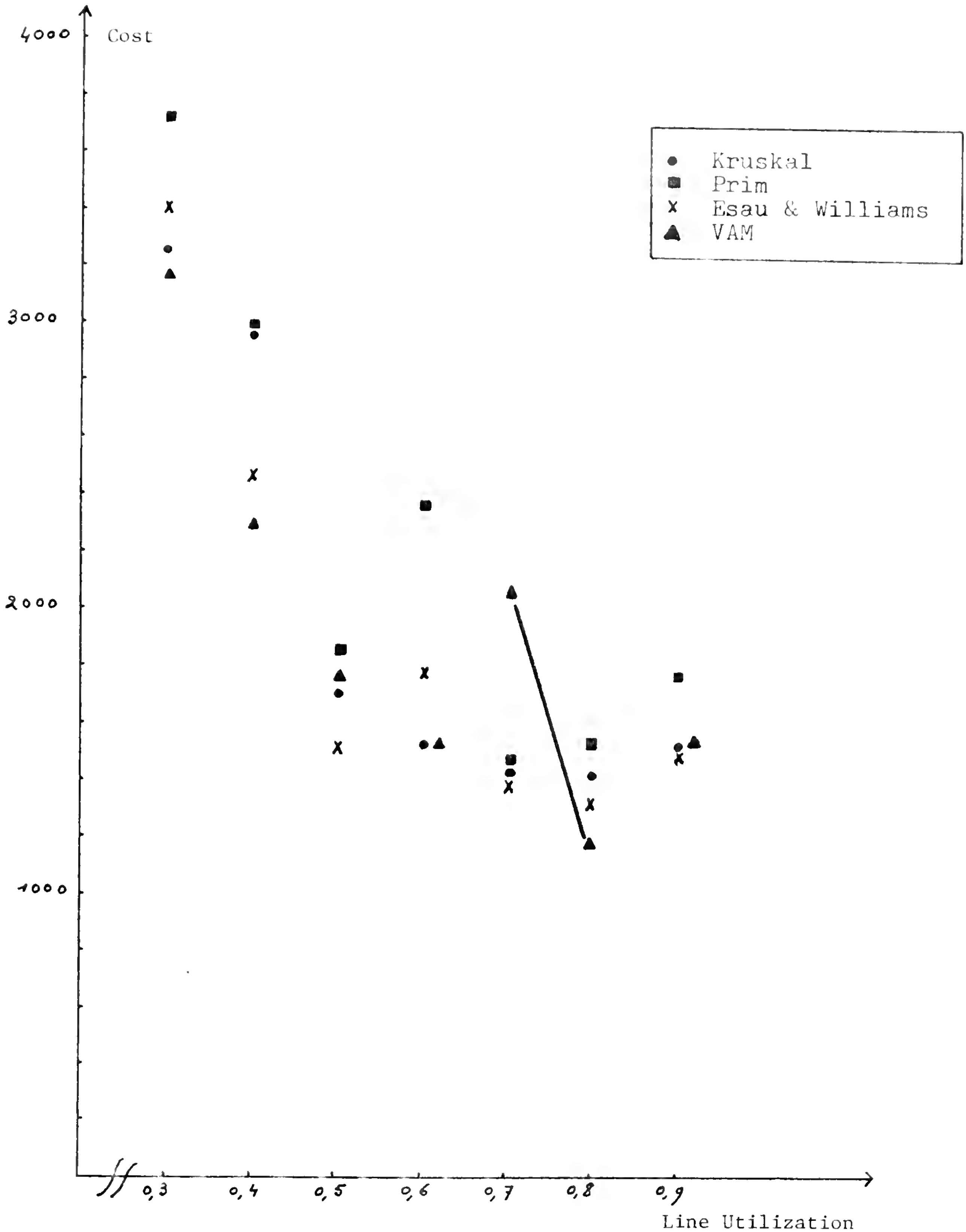

4000
3000
2000
1000
Cost
Kruskal
Prim
Esau & Williams
VAM
0,3
0,4
0,5
0,6
0,7
0,8
0,9
Line Utilization

Figure 7



4000
3000
2000
1000
Cost
Kruskal
Prim
Esau & Williams
VAM
0,3
0,4
0,5
0,6
0,7
0,8
0,9
Line Utilization

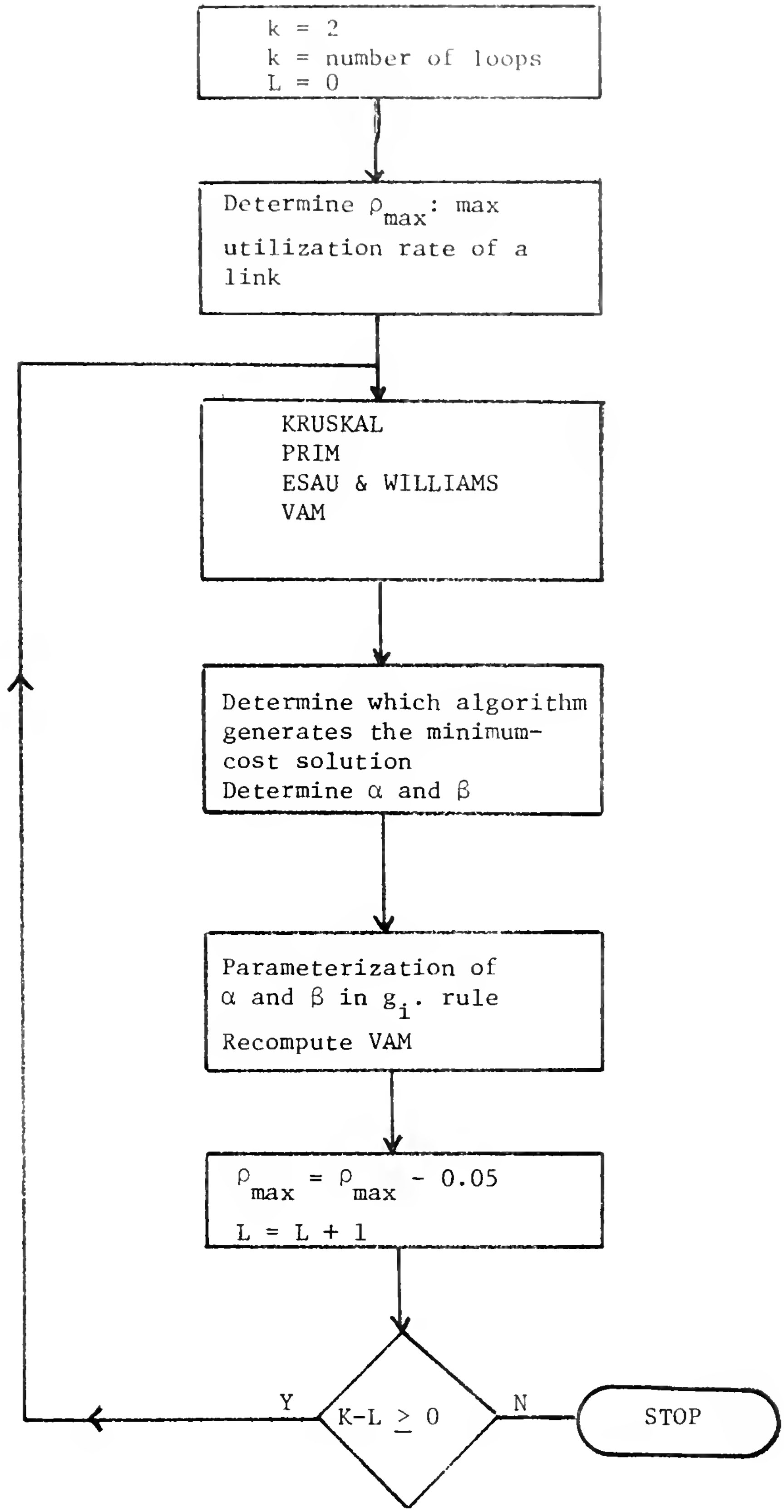


Figure 8: Multidrop design algorithm

We redefine g_i used in the VAM algorithm, to be: (6)

$$g_i = \alpha[\beta \times c_{i1} + (1-\beta) \times D_i]$$

where: $\alpha \geq 0$, $0 \leq \beta \leq 1$ constants

c_{i1} = cost of a direct link between node i and the central site (node 1)

$D_i = b_i - a_i$.

Note the very interesting properties of this rule:

	$\alpha = 1$	$\beta = 0$	generates VAM-algorithm
$g_i = 0$	$\alpha = 1$	$\beta = 0$	generates Esau/Williams
	$\alpha = 0$		generates Kruskal
$g_i = 0$	$\alpha = \infty$		generates Prim

According to the algorithm which has generated the minimum cost solution in Step 3 of the design procedure, these initial values will be used to do a parameterization of the g_i-rule. New values of α and β are generated until no further improvement can be achieved. At each step the VAM-algorithm is recomputed.

For the data used in Table I the solutions are recomputed using the model described above. The results are printed in Table III. In the first column the best solution of Table I is repeated.

Best Solution Table I	Solution using design algorithm	Improvement (percentage)
1814	1771	2.3
1189	1447	3.4
1025	1025	-
1725	1621	6
2923	2601	11
1365	1354	0.7
1848	1765	4.45
1246	1233	1.04
1740	1639	5.80
1762	1588	9.87
1833	1688	7.8
1932	1729	10.5
1437	1374	4.3
2370	2303	2.82
2014	1969	2.2
1083	1083	-
1868	1734	7.17
2021	1833	9.3
1954	1868	4.4
2011	1887	6.16

TABLE III

3.2 Two-level networks with concentrators

The problem of optimizing two-level networks with concentrators can be viewed as to determine: (i) the number of concentrators, (ii) the locations of the concentrators, (iii) the lay-out of the network, such that the overall network cost is minimized. In this working paper we shall assume that all concentrators are star-connected to the CPU, and have identical capacities. This assumption reduces (iii) to the determination of the line topology between terminals and concentrators.

The problem of optimizing a network with all terminals star-connected to either the optimally located concentrators or the CPU, can be formulated in a straightforward way:

Define: c_{ij} = cost of a direct link between node i and node j (N nodes)

Location of central computer is node 1.

$I_1; I_2; I_3; \ldots I_m$ are m possible concentrator locations $(m \leq N)$

G = cost of 1 concentrator.

The optimal solution will be indicated by the structure with the minimal cost ($D^* = \min_{j \varepsilon m} D_j$):

$$D_1 = \min_{I_1 \varepsilon N} \left[\sum_{k=2}^{N} \min \left(c_{k1}, c_{kI_1}\right)\right] + G$$

$$D_2 = \min_{I_1, I_2 \varepsilon N} \left[\sum_{k=2}^{N} \min \left(c_{k1}, c_{kI_1}, c_{kI_2}\right)\right] + 2G$$

$$\vdots$$

$$D_m = \min_{I_1, I_2, \ldots I_m \varepsilon N} \left[\sum_{k=2}^{N} \min \left(c_{k1}, c_{kI_1}, c_{kI_2}, \ldots c_{kI_m}\right)\right] + mG$$

The problem of optimizing two-level networks with terminals star-connected to the concentrators is almost analogous to the "Warehouse Location" problem of "Operations Research". Some authors propose exact methods for generating the mathematical optimum: (10), (11), (12), (13) and (14); others use heuristic techniques for solving large-scale problems: (18), (15), (16) and (17). In view of the impossibility of exact methods to handle problems of reasonable size, we decided to use heuristics for generating the solution.

The Kuehn & Hamburger algorithm (15) starts with no concentrators allocated to the network. Then, step by step concentrators are added based on the greatest cost savings which can be achieved. The Feldman/Lehrer/Ray algorithm (16) works in the opposite direction. Initially all nodes are concentrator locations; then step by step one of the concentrators is deleted until no more cost savings can be accomplished. Comparison of those two algorithms revealed that the Feldman/Lehrer/Ray algorithm generates slightly better results than the Kuehn & Hamburger heuristic. For large-scale problems, however, computing times for the Feldman/Lehrer/Ray program can be excessive. Although this is a constraint we recommend use of the Feldman/Lehrer/Ray algorithm in the optimization procedure.

When the terminals are multipoint connected to the concentrators the multipoint design procedure can be incorporated in the Feldman/Lehrer/Ray algorithm. However, for problems of reasonable size ($N > 100$, $I_m > 50$) computing times can be too excessive. A more feasible way has been described in (6), where after each deletion of a concentrator, terminals are re-allocated to new concentrators according to the link with the minimum cost, not violating the performance constraint.

3.3 Optimization of loop networks

The Vehicle Fleet Scheduling problems of Operations Research is analogous to the optimization problem of loop networks. The heuristic which can be used to determine a solution is due to Clarke and Wright (19).

CLARKE and WRIGHT Heuristic

[1] Initially all nodes are located on an individual loop (CPU → node → CPU);

[2] Find a pair of loops, resulting in the maximum cost savings when combined, and not violating the performance constraint. When cost savings are zero or negative: stop. Let's assume nodes i and j (located on loop i and j) generate the largest gain. The cost savings can be defined as:

$$c_{i1} + c_{j1} - c_{ij}.$$

[3] Add link i → j; delete links i → CPU and j → CPU; Return to [2].

Illustration: Figure 9.

4. Analysis

Although a performance measure has been used in the optimization heuristics, this does not guarantee that the response time of the network will be lower than the response time requirements. In the analysis section we have to determine the response times of the channels under various organization structures (polling or contention/half-duplex lines with either priority of input or priority of output versus full-duplex lines).

Contention systems with either half-duplex lines or full-duplex lines can be analyzed by simple queuing models (20), (6). The analysis of polling structures is almost identical to the analysis of a series of queues served in a cyclical way by one single server. Two possible disciplines can be distinguished:

- the server makes cyclical scans of every queue, and idles the consulted queue before addressing another queue;

Illustration: Loop Networks: Figure 9

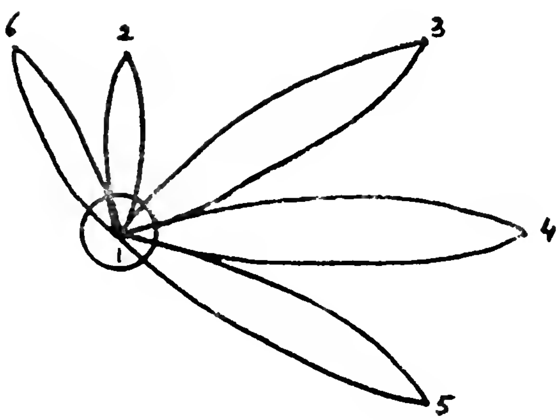
Cost matrix

	1	2	3	4	5	6
1	-	10	26	27	30	14
2	10	-	25	32	37	7
3	26	25	-	8	23	30
4	27	32	8	-	22	38
5	30	37	23	22	-	39
6	14	7	30	38	39	-

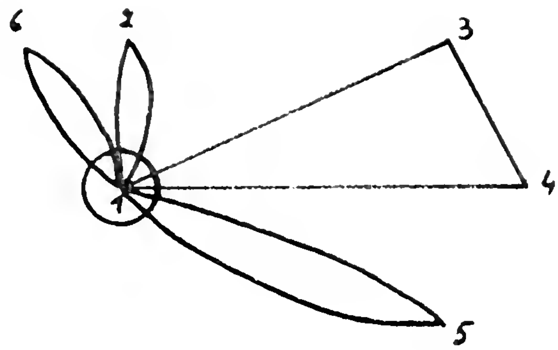
Max. Traffic on a loop: 255 bits/sec.

Number of bits from: (average per second)

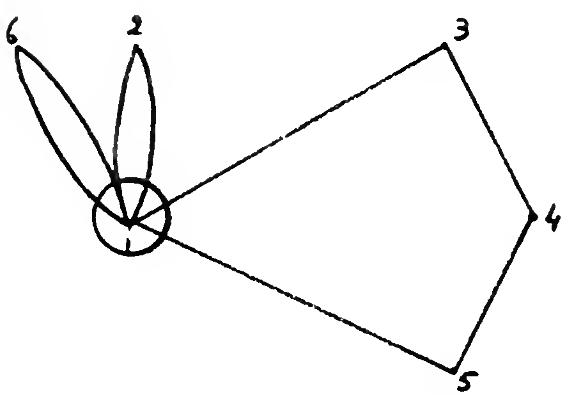
2	40
3	20
4	80
5	140
6	60



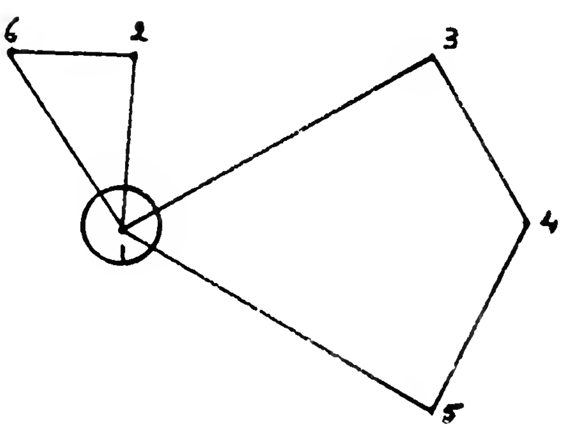
Iteration 1



Iteration 2



Iteration 3



Total cost: 101

- the server scans every queue, service, however, is limited to those messages which made up the queue at the arrival time of the server.

A lot of articles have been published on polling structures: (21), (22), (23), (24), (25). Chang (4) however, is the only one who analyses data communications networks with sophisticated hardware technologies (Terminal Control Units, Front-end Processors). Unless some minor changes, based on empirical results, this analysis technique for polling systems has been incorporated in the overall network procedure.

5. Evaluation

In the evaluation section of the procedure, the computed response times are compared with the response time objectives. When the computed response times exceed the requirement, the optimization has to be recomputed with a lower maximum performance ratio. When the response times are lower than the objective, the line capacities of the high-level network (between computer and concentrators) can easily be computed.

6. General Comments

The optimal solution for 30 small-size problems with 5 (125 possible trees) and 10 nodes (10^8 different trees) has been computed by means of an extensive enumeration method. The heuristic technique proposed in this working paper generated in about 85% of all cases a solution within 2% of the computed optimum. The maximum deviation observed between heuristic solution and optimum was 5.5%. The computing times for the proposed procedure were very reasonable.

In a forthcoming working paper the method will be extended to deal with STEINER trees and including research done on the reliability problem.

ACKNOWLEDGEMENTS

I would like to thank Professor Philippe Naert (Universiteit Antwerpen - UFSIA - Belgium) for his continuous interest and encouragement and for his many helpful suggestions. I wish to thank Professor Peter Chen (M.I.T., Alfred P. Sloan School of Management) for reviewing the first draft of this working paper, and Susan Wiseman for the fast and accurate typing.

REFERENCES

(1) Akoka, J., P. Chen, "A Framework for Distributed Information Systems", M.I.T. Sloan School of Management Working Paper.

(2) Abramson, N., F. F. Kuo, "Computer Communication Networks", New York, Prentice-Hall, 1973, 525 pages.

(3) Chandy, K. M., T. Lo, "The Capacitated Minimum Spanning Tree", University of Texas, Working Paper.

(4) Chang, J. H., "Terminal Response Times in Data Communications Systems", IBM Journal of Research and Development, May 1975, pp. 272-282.

(5) Prim, R. C., "Shortest Connection Networks and Some Generalizations", Bell System Technical Journal, 1966, Vol. 5, Nr. 3, pp. 142-147.

(6) De Backer, C., De Optimalisatie van Datacommunicatienetwerken", Doctoraal Proefschift, 1977.

(7) Kruskal, J. B., "On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem", Proc. Am. Math. Soc., 1956, Vol. 7, pp. 48-50.

(8) Esau, L. R., K. C. Williams, "A Method for Approximating the Optimal Network", IBM Systems Journal, 1966, Vol. 5, Nr. 3, pp. 142-147.

(9) Reinfeld, N. V., W. R. Vogel, Mathematical Programming, Englewood Cliffs, Prentic-Hall, 1958.

(10) Efroymson, M. A., T. L. Ray, "A Branch-Bound Algorithm for Plant Location", Operations Research, 1966, Vol. 14, pp. 361-368.

(11) Spielberg, K., "Algorithms for Simple Plant Location Problems with Some Side Conditions", Operations Research, January 1969, Vol. 17, pp. 85-111.

(12) Davis, P. S., T. L. Ray, "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem", Naval Research Logistics Quarterly, 1969, Vol. 16, pp. 331-344.

(13) Khumawala, B. M., "Warehouse Location Problem, Efficient Branch and Bound Algorithm", Management Science, Series B, August 1972, Vol. 18.

(14) Cooper, L., "Location-Allocation Problems", Operations Research, 1963, Vol. 11, Nr. 3, pp. 331-343.

(15) Kuehn, A. A., M. J. Hamburger, "A Heuristic Program for Locating Warehouses", Management Science, July 1963, Vol. 9, Nr. 4, pp. 643-666.

(16) Feldman, E., F. A. Lehrer, T. L. Ray, "Warehouse Location under Continuous Economics of Scale", Management Science, May 1966, Vol. 12, Nr. 9, pp. 670-684.

(17) Bahl, L. R., D. T. Tang, "Optimization of Concentrator Locations in Teleprocessing Networks", Proceedings of the Symposium on Computer-Communication Networks and Teletraffic, 1972, pp. 355-362.

(18) Marsten, R. E., "Approximate and Exact Solutions to a Class of Fixed Charge Problems", Northwestern University, Discussion Paper 10, 1972.

(19) Clarke, G., J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, Vol. 11, 1963, pp. 568-581.

(20) Martin, J., Systems Analysis for Data Transmission, Englewood Cliffs, Prentice-Hall, 1972.

(21) Leibowitz, M. A., "An Approximate Method for Treating a Class of Multiqueue Problems", IBM Journal of Research and Development, 1961, Vol. 5, Nr. 3, pp. 204-209.

(22) Leibowitz, M. A., "A Note on Some Fundamental Parameters of Multiqueue Systems", IBM Journal of Research and Development, October 1962, pp. 470-471.

(23) Hayes, J. F., D. N. Sherman, "A Study of Data Multiplexing Techniques", Bell System Technical Journal, Vol. 51, November 1972, pp. 1983-2011.

(24) Cooper, R. B., G. Murray, "Queues Served in Cyclic Order", Bell System Technical Journal, Vol. 48, 1969, pp. 675-690.

(25) Eisenberg, "Queues with Periodic Service and Changeover Time", Operations Research, 1972, Vol. 20, pp. 440-451.

Date Due

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

SEP 2 6 1986

MR 25 '88